

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-287064

(43)Date of publication of application : 31.10.1995

(51)Int.Cl.

G01S 7/285  
G01S 7/292  
G01S 13/66  
G06F 15/16

(21)Application number : 06-081548

(71)Applicant : MITSUBISHI ELECTRIC CORP

(22)Date of filing : 20.04.1994

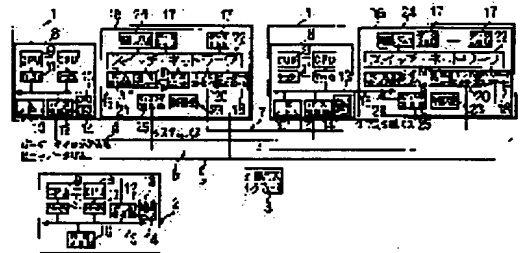
(72)Inventor : KANEKO TOMOMI

## (54) RADAR SIGNAL PROCESSOR

### (57)Abstract:

**PURPOSE:** To obtain a radar signal processor which is provided with an optimum cost-to-performance ratio, which can process a plurality of beams simultaneously and whose fault tolerance is high.

**CONSTITUTION:** A radar signal processor is provided with a plurality of CPUs 9 having cache memories 11 and with multi-CPU parts 8 having local buses 15 in which main memories have been mapped on memory spaces. It is provided with vector operation parts 16 in which a plurality of dedicated arithmetic units 19 have been connected by switch networks 22. They are connected, as clusters, by using buses between system buses and clusters. Thereby, when the plurality of CPUs 9 and the plurality of dedicated arithmetic units 19 are grouped individually in real time, effective resources are distributed optimally to a requested processing amount, and signals are processed. Since all the CPUs share the memory spaces, other CPUs can act for a certain CPU even when it is damaged.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平7-287064

(43)公開日 平成7年(1995)10月31日

(51)IntCl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 1 S	7/285	Z		
	7/292	A		
	13/66			
G 0 6 F	15/16	3 8 0 D		

審査請求 未請求 請求項の数5 O L (全 12 頁)

(21)出願番号 特願平6-81548

(22)出願日 平成6年(1994)4月20日

(71)出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72)発明者 金子 智巳

鎌倉市上町屋325番地 三菱電機株式会社

鎌倉製作所内

(74)代理人 弁理士 高田 守

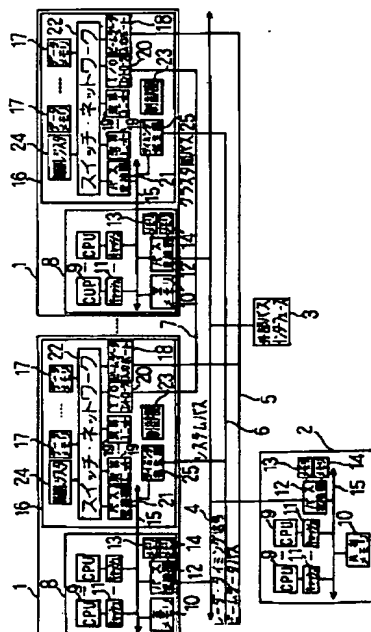
(54)【発明の名称】 レーダ信号処理装置

(57)【要約】

【目的】 最適な価格対性能比を有し、複数のビームを同時に処理可能なフォールトトレランス性の高いレーダ信号処理装置を得る。

【構成】 キャッシュメモリを備えた複数のCPUとメモリ空間上に主メモリがマッピングされているローカルバスを備えたマルチCPU部を備える。複数の専用演算ユニットをスイッチネットワークで接続したベクタ演算部を備える。これらをクラスタとしてシステムバス・クラスタ間バスを用いて接続した。

【効果】 複数のCPUと複数の専用演算ユニットをリアルタイムに各々グルーピングすることで、要求された処理量に対し有効資源を最適に分配し、信号処理を行う。全てのCPUがメモリ空間を共有することで、あるCPUが故障しても、他のCPUで代行することが可能である。



1

## 【特許請求の範囲】

【請求項1】 複数の信号処理プログラムモジュールからなるレーダ信号処理プログラムと制御プログラムが格納されている主メモリと、レーダ信号処理プログラムと制御プログラムを実行し、要求性能によって装備する数量を変更可能な複数のCPU (Central Processing Unit) と、前記主メモリのコピーを保持し、データ更新にともなう前記CPU間での時間的整合性を維持する機能を有し、前記CPUに対し最低1個以上有するキャッシュメモリと、前記CPUとシステムバス上に接続される他の機能ユニットとのデータ交換を制御する第1のバス変換器と、前記キャッシュメモリと前記主メモリと前記第1のバス変換器を接続し、データ内容整合性を維持する機能をバスプロトコル上で具備するローカルバスとから構成されるマルチCPU部を備え、複数の信号処理プログラムモジュールを複数のCPUでそれぞれ並列に実行するか、または、複数の信号処理プログラムモジュールのある1つの信号処理プログラムモジュールの処理対象データを複数のCPUで分割し並列実行することを前記CPUの一つであるスケジューラCPUで実行される有効資源割当てプログラムにより導き出された処理時間予測値に基づき、前記複数のCPUがそれぞれ処理すべき信号処理プログラムモジュールと処理対象データ数を要求されるタイムインターバル毎に毎回決定し、要求される信号処理モードを処理していくCPU負荷分散処理機能を有することを特徴とするレーダ信号処理装置。

【請求項2】 前記マルチCPU部と、前記マルチCPU部により制御され、前記マルチCPU部とのデータ交換を制御する第2のバス変換器と、入力信号データであるビームデータ、ベクタ演算中間データ及びベクタ演算結果を保持する機能を有し、要求性能によって装備する数量を変更可能な複数のデータメモリと、前記ビームデータが送信されてくるビームデータバスと、ビームデータをデータメモリに転送するビームデータ入力ポートと、ビームデータに対しベクタ演算を施し要求性能によって装備する数量を変更可能な複数の専用演算ユニットと、レーダタイミング信号を入力し専用演算ユニットへ演算開始信号を与えるとともに前記マルチCPU部のローカルバス上のメモリ空間に対し割込みメッセージ・コマンドとして前記マルチCPU部の任意のCPUに対し割込みを起動する機能を有するタイミング発生器と、前記データメモリ内のデータを外部と通信するためのI/Oコントローラと、前記データメモリと前記ビームデータ入力ポートと前記専用演算ユニットと前記第2のバス変換器を相互に接続しこれらの接続形態を動的に変更可能なスイッチネットワークと、前記マルチCPU部からパラメータ設定可能な制御情報を保持する制御レジスタと、制御レジスタの制御パラメータに基づきベクタ演算の制御を行う制御器とを備え、要求されるタイムインタ

2

ーバル毎に、前記マルチCPU部の1つであるスケジューラCPUで実行される有効資源割当てプログラムにより導き出された処理時間予測値に基づき、スイッチネットワークの接続形態を変更し、データメモリ、専用演算器、ビームデータ入力ポート、I/Oコントローラのグルーピング構成、グループ毎の信号処理メニュー及びビームデータ処理量を変え、要求される信号処理モードを処理する機能を有するベクタ演算部を有することを特徴とするレーダ信号処理装置。

【請求項3】 前記マルチCPU部と、前記ベクタ演算部から構成され、かつ、要求性能によって装備する数量を変更可能な複数のクラスタと、複数のクラスタ内のデータメモリ間で通信するためのクラスタ間バスと、前記マルチCPU部の機能を有し、全てのクラスタを制御するクラスタマスタと、外部装置とのデータ転送を制御する外部バスインターフェースと、前記クラスタと前記クラスタマスタと前記外部バスインターフェース間のデータ転送を行うためのシステムバスと、本装置内で利用可能な全ての資源が登録されている有効資源ライブラリを参照し、前記クラスタマスタ内のスケジューラCPUで実行される有効資源割当てプログラムより導かれた処理時間予測値に基づき、各クラスタのCPU、各クラスタのベクタ演算部、クラスタマスタのCPUの各資源に対し、資源割当てとスケジューリングを要求されるタイムインターバル毎に毎回行ない、要求される信号処理モードを処理していく機能を有することを特徴とするレーダ信号処理装置。

【請求項4】 前記マルチCPU部の前記ローカルバス上のメモリ空間に対し、レーダタイミング信号を変換し、信号処理の起動トリガを受信する毎に、前記主メモリのあるアドレスに対しセマフォフラグとして非分割の読出しバスサイクルとそれに引き続く書込みバスサイクルの排他的なアクセスを行うタイミング発生器を有することを特徴とする請求項2または、請求項3記載のレーダ信号処理装置。

【請求項5】 前記システムバス上に接続される全ての前記クラスタ内部の全ての主メモリが一つのメモリ空間上にマッピングされており、全てのクラスタ内部の全ての前記マルチCPU部のCPUからメモリ空間がアクセス可能であり、同一クラスタ内のキャッシュメモリが他のクラスタの共有メモリのデータをキャッシングしていることを示す情報を保持する第1の履歴管理メモリと、同一クラスタ内の主メモリのデータが最新データでない可能性があることを示すアドレス情報を保持する第2の履歴管理メモリと、ローカルバス、システムバス上に接続される他のクラスタ内部の全てのCPU間で全ての主メモリに対するデータ更新にともなう時間的整合性を維持する機能を有する前記マルチCPU部の第1のバス変換部と、全てのクラスタの第1のバス変換器を接続し、全ての主メモリに対するデータ更新にともなう時間的整

合性を維持する機能をバスプロトコルとして具備するシステムバスとから構成される請求項3記載のレーダ信号処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、複数ビームからなるデジタル受信データに対し、デジタル演算を施すことにより目標の搜索、追尾をリアルタイムに処理するレーダ信号処理装置の特に高速化、価格対性能比の向上、及びフォールトトレランスの向上に関するものである。

【0002】

【従来の技術】図7は、従来の信号処理装置の例である。図中、26はCPU部、27はベクタ演算部、3はレーダモード、信号処理メニューの受信及び信号処理結果の送信を外部装置間で行うための外部バスインターフェースである。4はCPU部26、ベクタ演算部27、外部バスインターフェース3とのデータ転送を行うためのシステムバスである。次に、CPU部26について説明する。10は複数の信号処理メニュープログラムが格納されている主メモリである。9はそのプログラムを逐次実行するCPUである。12はシステムバス4を介しCPU部26とベクタ演算部27、外部バスインターフェース3間でデータ転送を行うための第1のバス変換器である。15は主メモリ10、CPU9、バス変換器12間でデータ転送を行うためのローカルバスである。次に、ベクタ演算部27について説明する。5はビームデータが転送されるビームデータバス、6は信号処理の起動トリガ信号が送信されてくるレーダタイミング信号、18はビームデータを入力及びバッファリングを行うビーム入力ポートである。19はビーム入力ポート18から供給されるビームデータに対し、各々のユニット毎に各種の演算、例えばFIR (Finite Impulse Response) フィルタ演算、FFT (Fast Fourier Transform) 演算、CFAR (Constant False Alarm Rate) 演算を施す専用演算ユニットである。17はそれぞれの専用演算ユニット19の結果を保持するデータメモリである。23はベクタ演算部27の制御を行う制御部である。25はレーダタイミング信号6からの起動トリガを受信し、専用演算ユニット19に対し演算開始トリガ信号を発生し、また、CPU部26へシステムバス4の割込み専用線を用いてレーダタイミング信号6からの起動トリガを通知するためのタイミング発生器である。24は、専用演算ユニット19の動作モード、入力データ数などのベクタ演算部27の動作を制御する各種の制御情報を保有する制御レジスタである。21はCPU部26よりシステムバス4を介して、データメモリ17や制御レジスタ24をアクセスするための第2のバス変換器である。

【0003】次に従来のレーダ信号処理装置による動作について説明する。CPU9の実行すべき全ての信号処理

メニュープログラムは、主メモリ10にあらかじめ保持されている。外部バスインターフェースより受信した信号処理モードに従って、実行すべき信号処理プログラムを選択し実行する。CPU9は、制御レジスタ24に規定されたパラメータを設定するとともに、ベクタ演算部27の信号処理結果をデータメモリ17より読出し、目標の検出、追尾処理を行い、最終的な信号処理結果を外部バスインターフェース3に送出し、外部バスインターフェース3より外部装置へ転送する。また、ベクタ演算部27では、ビームデータバス5より受信したデータに対し、専用演算ユニット19を用いて、データメモリ17上に仮想的にマッピングされた空間上に、信号強度レベルがあるスレッシュホールドレベルより大きい空間的位置とその信号強度を格納する。これらの処理は、図8に示すように、信号処理メニュー毎に規定されたタイムインターバルを基準にCPU部26とベクタ演算部27をパイプラインステージとしたパイプライン処理により実行される。また、ベクタ演算部27内の各専用演算ユニット19においても、いくつかのレベルでパイプライン処理により実行される。CPU9とベクタ演算部27は、レーダタイミング信号6より受信した起動トリガによって同期がとられる。タイミング発生器25によりシステムバス4の割込み要求線によって割込みがCPU9に発生し、CPU9は、等価的に起動トリガを知ることができる。一方同時に、ベクタ演算部27では、タイミング発生器25により演算開始トリガが生成され、制御部23によって演算が開始される。

【0004】

【発明が解決しようとする課題】従来のレーダ信号処理装置は以上のように構成されているので、複数のビームを処理させる場合、レーダタイミング信号6のタイムインターバルを伸張させる必要がある。すなわち、ベクタ演算部27の専用ユニット19がパイプライン処理を行っているため、同時に複数ビーム分のデータを演算することができず、ビームを1本1本逐次処理させる必要があり、かつCPU9の演算性能の限界からも、規定のタイムインターバル内で処理することが不可能になる場合がある。タイムインターバルの伸張は、レーダ信号処理装置の重要な評価ファクタの1つであるレスポンスタイムの増加となり、好ましくない。しかしながら、CPU部26、ベクタ演算部27を対にし、これを複数ビーム本数分用意することで、タイムインターバルの伸張を回避可能であることが容易に想像される。複数の従来のレーダ信号処理装置を用意し、それぞれ独立に、ビームデータを与え、独立に制御することによって従来の信号処理を規定のタイムインターバル内で処理することが可能になる。この場合、複数のレーダ信号処理装置に対し、異なる信号処理メニューを実行させる場合、各レーダ信号処理装置において必要となる専用演算ユニット19の数量が必ずしも一定でないことから、専用演算ユニット19

5

稼働率は低下し、価格対性能比を下げることになる。また、運用時、CPU部26の故障、ベクタ演算部27の専用演算ユニット19、データメモリ17の故障は、レーダ信号処理装置全体の故障に結び付き、任務MTBF (Mean Time Between Fault) を下げることになる。

【0005】この発明は上記のような課題を解消するためになされたもので、タイムインターバルの伸張を回避し、かつ、ビーム数の増加に伴う必要となる資源の増加をおさえ、最適な価格対性能比を提供するとともに、資源の増加にともなう任務MTBFの低下をおさえることを可能にするレーダ信号処理装置を提供するものである。

【0006】

【課題を解決するための手段】この発明に係るレーダ信号処理装置は、ローカルバス上にバス変換器と、キャッシュメモリを介して複数のCPUを具備したマルチCPU部を備える。

【0007】また、データメモリ、専用演算ユニット、及びI/Oコントローラをスイッチネットワークで接続したベクタ演算部を備える。

【0008】また、これらマルチCPU部、ベクタ演算部を一つのクラスタとして、クラスタ間をクラスタ間バスで接続した複数のクラスタを備える。

【0009】また、レーダタイミング信号を変換し、セマフォフラグを発行するタイミング発生器を備える。

【0010】また、履歴管理メモリを備えたバス変換器と、データ更新にともなう時間的整合性を維持するバスプロトコルを具備したシステムバスを備える。

【0011】

【作用】この発明におけるレーダ信号処理装置は、複数ビーム分のデータを複数CPUによって並列に処理し、またベクタ演算部においてスイッチネットワークがデータメモリ、専用演算ユニットをビーム毎にグルーピングし複数ビーム分のデータを並列に処理する。

【0012】また、この発明におけるレーダ信号処理装置は、クラスタ間バスを用いてI/Oコントローラが、データメモリ間のデータ転送を行うことで、クラスタ間に分散したグルーピングを行い複数クラスタを用いて複数のビームを処理する。

【0013】また、この発明におけるレーダ信号処理装置は、タイミング発生器が割込みメッセージコマンド、又はセマフォの発行を行なうことにより、各クラスタのCPU、ベクタ演算部との同期を確保する。

【0014】また、この発明におけるレーダ信号処理装置は、キャッシュメモリ、バス変換器、履歴管理メモリが全てのCPU間で発生するデータ更新に伴う時間的整合性を確保する。

【0015】

【実施例】

実施例1. 以下、この発明の一実施例を第1図に示す。

6

図中、1は本装置内に複数あるクラスタ、2は各クラスタ1の処理内容を指示するクラスタマスタ、3は本装置と外部装置との通信を行う外部バスインターフェース、4はクラスタ1、クラスタマスタ2、外部バスインターフェース3間を接続し、データ転送を行うシステムバス、5は各クラスタ1にビームデータを分配するビームデータバス、6は各クラスタ1に入力されるレーダタイミング信号、7はクラスタ1間で局所的にデータ交換を行うためのクラスタ間バスである。8は複数の信号処理を並列に実行するマルチCPU部、9は信号処理プログラムを逐次実行するCPU、10はCPU9の信号処理プログラム、制御プログラム、又は、データを格納する主メモリ、11は基本的に主メモリ10のコピーを保持し、データとそのアドレス情報とデータをキャッシングしていることを4つのステートで示すいわゆるMESI (Modified, Exclusive, Shared, Invalid) 情報を各キャッシュライン毎に有するキャッシュメモリ、12はマルチCPU部の各CPUとシステムバスに接続された他の資源とのデータ転送を行うための第1のバス変換器、13はクラスタ1の各キャッシュメモリ11がキャッシングしている他のクラスタ1の主メモリ10にマッピングされているデータに対するアドレス情報とMESI情報を保持する第1の履歴管理メモリ、14は同一クラスタ内の主メモリ10のデータが最新データでない可能性のあるアドレス情報を保持する第2の履歴管理メモリ、15は主メモリ10、キャッシュメモリ11、第1のバス変換器12を接続し、データ転送を行うローカルバス、16はビームデータバス5または、クラスタ間バス7から入力されたデータに対しベクタ演算を施すベクタ演算部、17はビームデータ、ベクタ演算中間データを保持するデータメモリ、18はビームデータを入力し、データメモリ17へ転送するビーム入力ポート、19はデータメモリ17に格納されているビームデータ、ベクタ演算中間データに対し、各々のユニット毎に各種の演算、例えばFIR (Finite Impulse Response) フィルタ演算、FFT (Fast Fourier Transform) 演算、CFAR (Constant False Alarm Rate) 演算を施す専用演算ユニットである。20は他のクラスタ1との信号処理対象データをクラスタ間バス7を介してデータ交換を行うI/Oコントローラ、21はマルチCPU部8とベクタ演算部16間のデータ転送を行うための第2のバス変換器である。22はデータメモリ17と第2のバス変換器21、専用演算ユニット19、I/Oコントローラ20、ビーム入力ポート18間を接続し、その接続形態を変更可能なスイッチネットワーク、23はベクタ演算部16のベクタ演算制御を行う制御部、24は制御部の制御パラメータを保持する制御レジスタ、25はレーダタイミング信号6より受信した信号処理の起動トリガをベクタ演算部16の制御部23に与え、同時にCPU部8の各CPU9に対し、信号処理の起動タイミングを通

7

知するタイミング発生器である。

【0016】次に動作について説明する。マルチCPU 8部のCPU 9の実行すべき全ての信号処理プログラムと、制御プログラムは、主メモリ10にあらかじめ保持されている。CPU 9が主メモリ10上のプログラムやデータを読み出す場合、キャッシュメモリ11は、読み出しデータをキャッシングする。キャッシュメモリ11中に所望のプログラムやデータがあるときは、キャッシュメモリ11よりCPU 9に供給される。キャッシュメモリ11中に所望のプログラムやデータがない場合は、最新のデータを主メモリ10より読み出すか、相当アドレスの最新のデータを所有する他のキャッシュメモリ11より供給される。すなわち、キャッシュメモリ11には、一般的にバススヌーピング機能と呼ばれるローカルバス15の書き込みコマンド・読み出しコマンドをモニタする機能がある。なお、各クラスタ1の主メモリ10と各クラスマスタ2の主メモリ10は、各クラスタ1の第1のバス変換器12のアドレスデコード、各主メモリ10のアドレスデコードに従って1つのメモリ空間上に図2のようにマッピングされている。第1のバス変換器12に付属する2種類の履歴管理メモリがある。第1の履歴管理メモリ13は、そのマルチCPU部8内のキャッシュメモリ11が他のマルチCPU部8の主メモリ10のデータをキャッシングしていることを4つのステートで示すMESI情報とそのアドレス情報を保持する履歴管理メモリである。第2の履歴管理メモリ14は、同一クラスタ1内の主メモリ10のデータが最新データでない可能性があることを示すアドレス情報を保持する履歴管理メモリである。次にこれら履歴管理メモリ13、14により、本レーダ信号処理装置内の各CPU 9間で最新更新データの授受を保証する実行例について説明する。例えば、複数のクラスタ1のいくつかのCPU 9のキャッシュメモリ11が同一アドレスのデータをキャッシングしているときに、あるCPU 9が異なるクラスタ(1)の主メモリ10にマッピングされているそのアドレスに対してデータ書き込み操作を行い、他のクラスタ1のCPU 9がそのアドレスに対して読み出し操作を行う場合について説明する。あるCPU 9が書き込み操作を行う以前では、その書き込みアドレスがマッピングされている主メモリ10の同一クラスタ1内の第2の履歴管理メモリ14にはこのアドレスは登録されておらず、主メモリ10のデータは、最新データであることが示されている。同一アドレスの同一データをキャッシングしているキャッシュメモリ11内のそのアドレスに対応するキャッシュラインのステータスを示すいわゆるMESI情報は、Sharedを示しており、書き込み操作を行うCPU 9のキャッシュメモリ11も同一アドレスの同一データをキャッシングしている場合には、同様にSharedを示している。キャッシングされていない場合は、キャッシュメモリ11のキャッシュラインは、Invalidを示

8

しているか、アドレス情報として登録されていない。このようなキャッシュ状態にある場合、あるCPU 9が書き込み操作を行うと、そのCPU 9のキャッシュメモリ11は、書き込みデータをキャッシングするとともに、キャッシュラインのステータスは事前のステータスに従って、Shared又は、InvalidからExclusiveに状態が遷移する。同時に、ローカルバス15上に書き込みコマンドサイクルを発行する。同一クラスタ内のキャッシュメモリ11にそのアドレスに対応するデータをキャッシングしていた場合には、バススヌーピング機構が動作し、相当するキャッシュラインのステータスは、SharedからInvalidに状態が遷移する。また、第1のバス変換器12も同様にローカルバス15を監視するバススヌーピング機構が動作し、第1の履歴管理メモリ13内を検索することによって、その書き込みコマンドサイクルのアドレスは、他のクラスタ1の主メモリ10のデータであることを4つのステートで示すMESI情報がSharedを示していることから認識され、システムバス4上にこの書き込みコマンドサイクルを発行するとともに、このアドレスに対応するステータスをSharedからExclusiveに状態を遷移させる。次に、システムバス4上に発行された書き込みコマンドサイクルは、各クラスタ1の第1のバス変換器12のバススヌーピング機構によって、各々の第1の履歴管理メモリ13内を検索することによって、そのアドレスに対応するMESI情報がSharedを示している各クラスタ1の第1のバス変換器12は、そのクラスタ内のキャッシュメモリ11に古いデータをキャッシングしていることを認識し、第1のバス変換器12は、第1の履歴管理メモリ13のそのアドレスに対応するMESI情報をSharedからInvalidに状態を遷移させるとともに、そのクラスタ1内のローカルバス15上にも書き込みコマンドサイクルを発行することで、同様にして、そのクラスタ1内の各キャッシュメモリ11の相当するキャッシュラインのステータスも、SharedからInvalidに状態を遷移させる。一方、システムバス4上に発行された書き込みコマンドサイクルは、本来、主メモリ10がマッピングされているクラスタ1の第1のバス変換器12のアドレスデコードによって認識され、ローカルバス15上にも書き込みコマンドサイクルを発行することで主メモリ10に書き込み操作を行うとともに、第2の履歴管理メモリ14に主メモリ10上のデータは、以後最新データでない可能性があるものとしてこのアドレスを登録する。その後、システムバス4の書き込みコマンドサイクルを終了し、さらに書き込みコマンドサイクルを発行したCPU 9のクラスタ1内のローカルバス15の書き込みコマンドサイクルも終了する。このようにして、書き込みコマンドを発行したCPU 9のキャッシュメモリ11の相当アドレスのキャッシュステータスとそのクラスタ1の第1の履歴管理メモリ13のステ

タスがExclusiveを示し、他の全てのキャッシュメモリ11と第1の履歴管理メモリ13のステータスはInvalidを示すことによって、このアドレスに対応するデータの最新データの格納場所が特定される。次にこの時点で、他のクラスタ1のCPU9が同一アドレスに対し読出しコマンドを発行した場合には、各キャッシュメモリのバスヌーピング機構と、第1の各履歴管理メモリ13を参照メモリとして第1のバス変換器12のバスヌーピング機構が動作し、Exclusiveを示していたキャッシュメモリ11の相当アドレスのキャッシュステータスとそのクラスタ1の第1の履歴管理メモリ13のステータスは、Sharedに遷移するとともに、読み出しコマンドを発行したCPU9は、キャッシュメモリ11の相当アドレスのキャッシュステータスがInvalidであることと、前回書込みコマンドを発行したCPU9のそのクラスタ1内の第1の履歴管理メモリ13のステータスがExclusiveを示していたことから、そのアドレスがマッピングされている主メモリ10のデータは最新であると判断し、主メモリ10がマッピングされているクラスタ1の第1のバス変換器12、ローカルバス15を介して、主メモリ10から読み込み操作を行い、読出しコマンドを発行したCPU9のキャッシュメモリ11にキャッシングするとともに、キャッシュメモリ11の相当アドレスのキャッシュステータスとそのクラスタ1の第1の履歴管理メモリ13のステータスも、Sharedに遷移させる。また、書込みコマンドを発行したCPU9のキャッシュメモリ11の相当アドレスのキャッシュステータスとそのクラスタ1の履歴管理メモリ13のステータスがExclusiveを示し、他の全てのキャッシュメモリ11と履歴管理メモリ13のステータスがInvalidを示している状態において、書込みコマンドサイクルを発行したCPU9がさらに続けて同一アドレスに対して書込みコマンドを発行した場合には、書込みコマンドを発行したCPU9のキャッシュメモリ11の相当アドレスに対するキャッシュステータスとそのクラスタ1の履歴管理メモリ13のステータスを、以上説明した方法と同様にして双方ともModifiedにする。このとき、システムバス4上には、書込みコマンドは発行されず、ローカルバス15上のみ書込みコマンドは発行され、他のクラスタ1上のキャッシュステータスと履歴管理メモリ13のステータスは変化しない。同一のCPU9が、さらに同一アドレスに対して書込みコマンドを発行する場合は、このキャッシュステータスと履歴管理メモリ13のステータスは維持される。ここで、書込みコマンドを発行したアドレスがマッピングされている主メモリ10と同一のクラスタ1のCPU9がこのアドレスに対して読出しコマンドを発行した場合、主メモリ10から古いデータを読み出さないように、各キャッシュメモリ11と第1のバス変換器12は以下のように動作す

る。すなわち、CPU9がこのアドレスに対して読出しコマンドを発行すると、当然このCPU9のキャッシュメモリ11は、Invalidであるのでローカルバス15に読出しコマンドが発行される。ローカルバス15には、このアドレスがマッピングされている主メモリ10があるので主メモリ10がこの読出しコマンドに応答しようと試みるが、ここで、第1のバス変換器12が、バスヌーピング機構によって、第2の履歴管理メモリ14に登録されているアドレスに対する読出しコマンドであることを検出し、第1のバス変換器12は、読出しコマンドを発行したキャッシュメモリ11に対しリトライ信号を返送し、ローカルバス15の使用権を得てキャッシュメモリ11とCPU9の動作を凍結する。と同時に、システムバス4上に読出しコマンドを発行する。最新データを保持しているクラスタ1のキャッシュステータスと第1の履歴管理メモリ13のステータスは双方ともModifiedになっており、最新データを保持しているクラスタ1の第1のバス変換器12は、バスヌーピング機構によって、今度は、双方ともSharedに遷移するとともに、最新データを読出しコマンドに対するデータとして供給する。最新データを受信した第1のバス変換器12は、最新データを主メモリ10に書込むとともに、第2の履歴管理メモリ14上に登録されているこのアドレスに情報を削除し、主メモリ10が最新データを保持していることを示す。さらにこの第1のバス変換器12は、ローカルバス15の使用権を解放してキャッシュメモリ11とCPU9の動作凍結を解除する。これによって、キャッシュメモリ11が再度ローカルバス15上に読出しコマンドを発行し、主メモリ10が最新データを供給して、キャッシュメモリ11にキャッシングするとともにキャッシュステータスをSharedとする。さらに、CPU9は、キャッシングされた最新データをキャッシュメモリ11から受け取る。このように動作することで、各CPU9間で最新更新データの授受が保証され、各CPU9からは主メモリ11に対し、データ更新の時間的連続性、アドレス空間の対象性を有している。このようなマルチCPU部8で、複数の処理を実行する場合、代表的な処理方法として、図3に示す処理単位による分割方法と図4に示すデータ数による分割方法の2種類の処理方法があり、また通常はこれらの折中方法で実行される。処理単位による分割方法は、各処理において相互にデータ依存性のない複数の異なる処理を複数のCPU9で並列に処理する方法である。一方データ数による分割方法は、処理対象となるデータ数をCPU9の個数で分割し、異なるデータ群に対して同一の処理を全てのCPU9で実行する方法である。マルチCPU部8の主メモリ10に格納されている信号処理プログラムは、各信号処理メニュー毎にモジュール化されており、その信号処理メニュープログラムモジュールは処理対象となるデータ数を任意に分割実行可能なように



構造化されている。したがって、信号処理プログラムは、相互にデータ依存性のない各信号処理メニュープログラムに対しては、待機状態にある複数のCPU9で同時に並列に実行し、また、信号処理メニュー間で相互にデータ依存性がある場合には、待機状態にある複数のCPU9で、同一の信号処理メニュープログラムを処理対象となるデータ数を分割して実行する。クラスタマスタ2のマルチCPU部8、各クラスタ1のマルチCPU部8、各クラスタ1のベクタ演算部16の実行制御に関する相互同期は、従来の装置と同様に、信号処理モード毎に規定されたレーダタイミングから受信されるタイムインターバルを基準に、クラスタマスタ2のマルチCPU部8、各クラスタ1のマルチCPU部8、各クラスタ1のベクタ演算部16をそれぞれパイプラインステージとしてパイプライン処理により実行される。レーダタイミング信号6からの起動トリガの受信とともに、各クラスタ1のうちのある1つのクラスタ1のタイミング発生器25は、同一クラスタ1内のローカルバス15上に割込みメッセージコマンドを発行し、この割込みメッセージコマンドは、同一クラスタ1内の第1のバス変換器12によってシステムバス上にも発行される。クラスタマスタ2の全てのCPU9は、この割込みに応答するが、クラスタマスタ2のCPU9のうちある1つのCPU9が割込み処理の実行権を得る。クラスタ2の割込み処理の実行権を得たCPU9は、以後クラスタマスタ2のスケジューラとして動作し、クラスタマスタ2の主メモリ10に格納されている制御プログラムにタスクスイッチし、図5に示すようなフローチャートに従ってレーダ信号処理装置の有効資源割り当てとスケジューリングの決定が行われる。外部バスインターフェース3より外部装置が次のタイムインターバルで実行すべき要求処理量、すなわちビーム本数、各ビームの信号処理メニュー、各ビームのデータ数、管理ビームの処理優先度、各ビームに対する信号処理メニューの処理優先度、完了時間を獲得する。このとき、このCPU9は、クラスタマスタ2の主メモリ10上に構築されている有効資源ライブラリと要求処理量とのつき合わせを行う。有効資源ライブラリは、一種のデータベースであり、本レーダ信号処理装置内で利用可能な全ての資源が登録されている。具体的には、クラスタマスタ2のCPU数、クラスタ数、各クラスタ1のCPU数、各クラスタ1のデータメモリ数、各クラスタ1の専用演算ユニットタイプ及び数量等が登録されており、パワーオンスタート時に各資源の有無を確認後、利用可能になる。また、有効資源に故障が発見された場合には、この有効資源ライブラリより削除される。CPU9は、次のタイムインターバルで実行すべき要求処理量を分析し、要求処理を最大限に消化するよう規定の方針に従って有効資源ライブラリに登録されている有効資源の資源割当てと処理のスケジュールの決定を行う。その情報は、制御パラメータとして、クラスタマ

スタ2の第1のバス変換器12、システムバス4、各クラスタ1の第1のバス変換器12、各クラスタ1のローカルバス15を介して各クラスタ1の主メモリ10に転送される。その後、割込みメッセージコマンドを制御パラメータの転送と同様にして各クラスタ1の各CPU9に発行し、制御パラメータを転送したことを通知する。また、クラスタマスタ2の他のCPU9は、ループ処理を行って実行可能状態を示すクラスタマスタ1の主メモリ10上に確保されたセマフォをテストし待機状態に入る。スケジューラのCPU9は、クラスタマスタ1の各CPU1で行う処理をクラスタマスタ1の主メモリ10上に制御パラメータを書き込み後、セマフォに対して実行可能状態を示すマーキングを行う。待機状態にあったCPU9は、セマフォを獲得することでループ処理を抜け出し、制御パラメータで指定された所定の処理を開始する。所定の処理の終了時、スケジューラのCPU9が、実行完了状態を示すセマフォを、起動時と同様に他のCPU9から受け取ることでクラスタマスタ1内の処理が完了することを知る。一方、各クラスタ1に制御パラメータが発行されたのち、後続の割込みメッセージコマンドによってクラスタ1内の全てのCPU9は、この割込みに対し応答するが、ある1つのCPU9だけが、割込み処理の実行権を得る。クラスタ1の割込み処理の実行権を得たCPU9は、以後クラスタ1のスケジューラとして動作し、クラスタ1の主メモリ10に格納されている制御プログラムにタスクスイッチし、クラスタ1内の他のCPU9に対し、クラスタ1の主メモリ10に制御パラメータとして書き込み後、クラスタマスタ2のCPU9間のセマフォ授受と同様にして、制御パラメータ情報に基づく所定の処理の起動及び終了通知を行う。また、クラスタ1のスケジューラとなったCPU9は、クラスタマスタ2から転送された制御パラメータ情報に基づき、ベクタ演算部16の制御レジスタ24へ、ローカルバス15、第2のバス変換器21、スイッチネットワーク22を介して、制御データを転送する。この制御データには、ベクタ演算部16を制御するための情報が含まれており、専用演算ユニット19の、ビーム及び処理メニュー毎のグルーピングと演算モード、データメモリ17のビーム及び処理メニュー毎のグルーピングとモード、ビームデータ入力ポート18のビームデータバス5から取得するビーム番号、ビームデータ数、I/Oコントローラ20の処理メニュー毎のモード及び転送データ数、スイッチネットワーク22のビーム及び処理メニュー毎のネットワーク形態である。この制御データが制御レジスタ24に格納された後、タイミング発生器25によって、レーダタイミング6の起動トリガが検出されると、制御部23は、制御データの制御情報に従って、ベクタ演算部16の各資源を制御し、信号処理メニューを実行していく。図6は、この処理の様子を示したものであり、異なるビームに対し、異なる信号処理メニューを

それぞれグルーピングした専用演算ユニット19とデータメモリ17を用いて実行し、ビームデータ入力ポート18からは、同時に次のタイムインターバルで処理するビームデータを複数ビーム分取り込み、データメモリ17に転送する。なお、グルーピングが、クラスタ1間をまたがる場合は、I/Oコントローラ20、クラスタ間バス7を介してデータメモリ17間でデータ転送を行い、ベクタ演算中間データ、ビームデータの交換を行う。このようにして、ベクタ演算部16において、信号処理メニューが実行され、目標であると想定される信号強度レベルがあるスレッシュホールドレベルより大きい空間的位置とその信号強度レベルデータメモリ17に仮想的にマッピングされたメモリ空間上に格納される。ベクタ演算部16の処理終了を検出したタイミング発生器25は、ベクタ演算部16の処理終了を同一クラスタ1内のマルチCPU部8のスケジューラのCPU9へ通知するため割り込みメッセージがコマンドをローカルバス15上に発行する。次にスケジューラのCPU9は、割り込みを受け付け、データメモリ17よりスイッチネットワーク22、第2のバス変換器21、ローカルバス15を介して、目標であると想定される信号強度レベルがあるスレッシュホールドレベルより大きい空間的位置とその信号強度レベルを読出し、主メモリ10へ転送する。さらに、スケジューラのCPU9は、待機状態にある他のCPU9に対し、セマフォの授受によって、ビーム間に独立した目標の判定処理、及び、ビーム間の相関処理の起動を行う。このとき、ビーム間の相関処理において他のクラスタ1または、クラスタマスタ2の主メモリに所望のデータが存在する場合には、ローカルバス15、第1のバス変換器12、システムバス4を介して他のクラスタ1又は、クラスタマスタ2の主メモリ10に読出しコマンドを発行することにより所望のデータの授受を行う。これらのビーム間に独立した目標の判定処理、及びビーム間の相関処理は、いくつかの信号処理メニュープログラムモジュールから構成されており、待機状態にあるCPU9を幾つかのCPU9毎にグルーピングすることにより、グループ単位では、異なったデータ依存性のない並列実行可能な信号処理メニュープログラムモジュールを並列実行し、同一グループ内のCPU9は、処理対象データを分割し、同一の信号処理メニュープログラムを並列実行する。これらのビーム間に独立した目標の判定処理、及び、ビーム間の相関処理が終了すると、スケジューラのCPU9と他のCPU9間で、セマフォの授受をすることによって、スケジューラのCPU9に処理終了を通知する。さらに、スケジューラのCPU9はクラスタ1の全ての信号処理が終了したことをクラスタマスタ2に通知するため、割り込みメッセージコマンドをシステムバス4上に発行し、クラスタマスタ2はこれに応答する。すなわち、クラスタマスタ2では、割り込み処理の実行権を獲得したスケジューラのCPU9が、他のCPU

9に対し、追尾処理、状況認識処理のセマフォの授受を用いて起動を行う。他のCPU9は、各クラスタ1の主メモリ10から、ローカルバス15、第1のバス変換器12、システムバス4を介して、目標情報、状況データを取得し、追尾処理、状況認識処理を行う。これらの追尾処理、状況認識処理は、いくつかの信号処理メニュープログラムモジュールから構成されており、待機状態にあるCPU9のいくつかのCPU9毎にグルーピングすることにより、グループ単位では、異なったデータ依存性のない並列実行可能な信号処理メニュープログラムモジュールを並列実行し、同一グループ内のCPU9は、処理対象データを分割し、同一の信号処理メニュープログラムを並列実行する。これらの追尾処理、状況認識処理が終了すると、追尾処理、状況認識処理の演算結果である最終ターゲットファイルを外部バスインターフェース3へ送出し、外部装置へ転送する。以上の処理をタイムインターバルを基準として毎回繰り返すことにより、所望の信号処理を実行する。

【0017】実施例2. 次にこの発明の第2の実施例を説明する。従来のレーダ信号処理装置では、タイムインターバルの基準時刻を通知する手段としてシステムバス4の割り込み専用線をタイミング発生器25が駆動する方法を用いていた。また、実施例1において、マルチCPU部8の各CPU9にタイムインターバルの基準時刻を通知する手段として、タイミング発生器25は、レーダタイミング信号6を受信し、マルチCPU部8のローカルバス15上に割り込みメッセージコマンドを発行することで実現していた。この場合、割り込みを受け付ける各CPU9は、一時的に現在まで実行していた処理を中断し、処理の切り替え操作、具体的には、現在まで実行していた処理のタスクコントロールブロックを主メモリ10上のスタック空間に退避する操作と割り込みメッセージコマンドの解析と新たに実行すべき処理のタスクコントロールブロックの初期化を実行する必要がある。これら一連の割り込み処理を回避する手段として実施例2を説明する。実施例2の構成は、実施例1の構成とタイミング発生器25の機能を除き、全く同じものである。主メモリ10の特定アドレスをセマフォ授受のアドレスとし、タイミング発生器25は、レーダタイミング信号6から起動トリガ信号を受信すると、その特定アドレスに対し、ローカルバス15上に読出しコマンドを発行する。さらに、読出しデータに対し、事前に規定されたビット位置にセマフォフラグとして"1"を論理和をとってセットし、これを書込みデータとして再度特定アドレスに対し、ローカルバス15上に書込みコマンドを発行する。これら一連の読出しコマンド、書込みコマンドは、非分割の連続サイクルとして発行され、ローカルバス15のプロトコルの一種として、この一連の非分割の連続サイクルが分割されることがないことを保証する。マルチCPU部8は、待機状態にあるとき、常にこのセマ

オフラグに対し、連続的に読出しコマンドを発行する。読出しコマンドは、一度主メモリ10から読出しデータが供給されるが、以降はキャッシュメモリ11に対して読出しが行われる。この処理は、セマフォフラグが“1”にセットされるまで、全ての待機状態のCPU9で続行される。ここで、タイミング発生器25がセマフォフラグのビット位置に“1”をセットする書き込みコマンドをローカルバス15に発行すると、主メモリ10の特定アドレスに書き込み操作が行われるとともに、キャッシュメモリ11のバススヌープ機構が動作し、CPU9が発行する次の読出しコマンドは、キャッシュメモリ11内の古いデータではなく、主メモリ10の最新更新データすなわちセマフォフラグがCPU9に供給される。待機状態にあるCPU9の幾つかは、セマフォフラグがセットされたものと判断し、タイミング発生器25と同様にセマフォフラグ読出しコマンドとセマフォフラグをリセットするすなわち“0”を書き込むための書き込みコマンドを非分割の連続サイクルとして発行する。しかしながら、待機状態にあるいくつかのCPU9が、この処理をほぼ同時に行うので、ローカルバス15のバス調停回路によって選択された唯一のCPU9がセマフォフラグを獲得、すなわちセマフォ読出しコマンドにおいて“1”を検出することができる。他の待機状態にあるいくつかのCPU9は、セマフォを獲得することはできない。セマフォを獲得した唯一のCPU9は、その後スケジューラとして実施例1と同様の動作を行う。なお、クラスタ1の場合と同様に、クラスタマスタ2においても特定のクラスタ1のタイミング発生器25がクラスタ1のローカルバス15、クラスタ1の第1のバス変換器12、システムバス4、クラスタマスタ2の第1のバス変換器12、クラスタマスタ2のローカルバス15を介して、クラスタマスタ2の主メモリ10にセマフォフラグの発行を行うことによってスケジューラとなるCPU9を決定し、以後の処理は、実施例1と同様に動作が行われる。

#### 【0018】

【発明の効果】この発明は、以上のように構成されているため、以下に記載されるような効果を有する。一般に、複数のビームに対し、異なる信号処理メニューを実行させる頻度が高いことから、必ずしもビーム1本の最大要求処理量に見合うCPU、専用演算器等の資源量を、ビーム本数分同数用意する必要はなく、少ないビーム数や、少ない要求処理量に対する信号処理モードでは、タイムインターバルを短くすることが可能であり、レーダ信号処理装置の重要な評価ファクタの一つであるレスポンスタイムを短くすることが可能になり、追尾精度、目標検出精度の向上が期待できる。

【0019】また、信号処理プログラムを変更することなく、実装時に有効資源ライブラリに数量を登録するだけで、CPU負荷分散処理、スイッチネットワークによ

る柔軟なグルーピングによって、専用演算器、CPUの稼働率を従来よりも上げることが可能であり、最適な価格対性能比を持つレーダ信号処理器が提供可能になる。

【0020】また、システムバス、ローカルバスに接続されるバススヌーピング機構を備えたキャッシュメモリ、バス変換器によって、複数のクラスタの複数のCPU間で発生するデータ更新にともなう時間的整合性を維持する機能が実現されていることから、システムバス、ローカルバス上のバス転送頻度を軽減することが可能になり、バス競合にともなうCPU演算時間の増大を抑えることが可能になり、結果的により大きな要求処理に対応することが可能になる。

【0021】また、複数のクラスタの複数のCPU間で発生するデータ更新にともなう時間的整合性を維持する機能が実現されていることから、いくつかのCPUが故障した場合も、有効資源ライブラリより故障したユニットを削除することだけで、他のクラスタ、クラスタマスタのCPUを用いてグルーピングすることが可能になり、部分的な故障が発生した場合も演算性能を極力落とさずに継続処理可能なことと、CPUの故障、専用演算器の故障、クラスタの故障ときめ細かい故障分離が可能な高いフォールトトレランス性を有するレーダ信号処理装置が提供できる。

【0022】また、レーダタイミングをCPUに通知する場合、タイミング発生器によるセマフォの発行と、CPUのポーリング処理によるタスクスイッチという方法を用いているため、割込みを使用する従来の装置にくらべ、短い時間でタスクスイッチを行うことが可能になり、結果的により大きな要求処理に対応することが可能になる。

【0023】また、バススヌーピング機構を備えたキャッシュメモリ、バス変換器によって、複数のCPU間で発生するデータ更新に伴う時間的整合性を維持する機能が実現されていることから、CPU間で主メモリを介したデータの受け渡しを行うプログラム作成が容易になる。

#### 【図面の簡単な説明】

【図1】この発明の実施例を示すブロック図である。

【図2】CPUから見たメモリ空間のメモリマップを示す図である。

【図3】複数のCPUが処理を分割して並列実行する様子を示した図である。

【図4】複数のCPUがデータ数を分割して並列実行する様子を示した図である。

【図5】有効資源割当てプログラムのフローチャートである。

【図6】ベクタ演算部のグルーピングを行った1例を示した図である。

【図7】従来のレーダ信号処理装置を示す図である。

【図8】従来のレーダ信号処理装置の処理タイミングを

17

示すタイムチャートである。

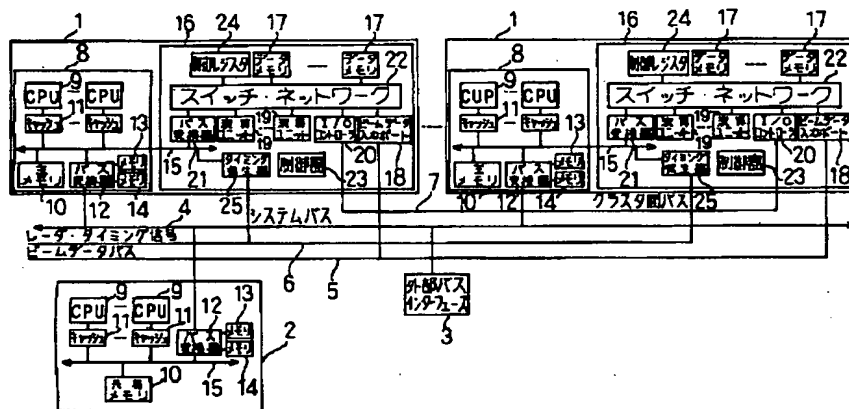
【符号の説明】

- 1 クラスタ
- 2 クラスタマスタ
- 3 外部バスインターフェース
- 4 システムバス
- 5 ビームデータバス
- 6 レーダタイミング信号
- 7 クラスタ間バス
- 8 マルチCPU部
- 9 CPU
- 10 主メモリ
- 11 キャッシュメモリ
- 12 第1のバス変換器

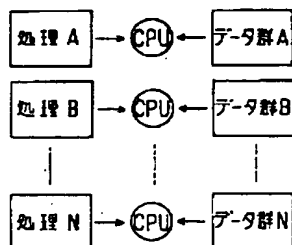
18

- 13 第1の履歴管理メモリ
- 14 第2の履歴管理メモリ
- 15 ローカルバス
- 16 ベクタ演算部
- 17 データメモリ
- 18 ビームデータ入力ポート
- 19 専用演算ユニット
- 20 I/Oコントローラ
- 21 第2のバス変換器
- 10 22 スイッチネットワーク
- 23 制御部
- 24 制御レジスタ
- 25 タイミング発生器

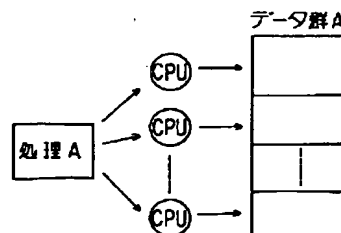
【図1】



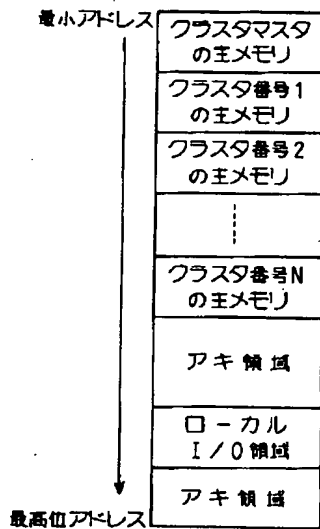
【図3】



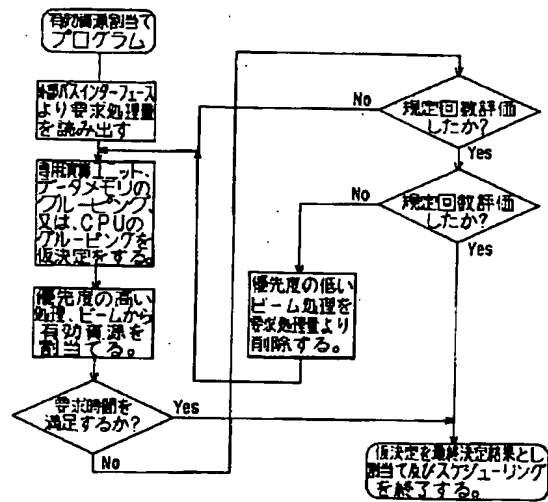
【図4】



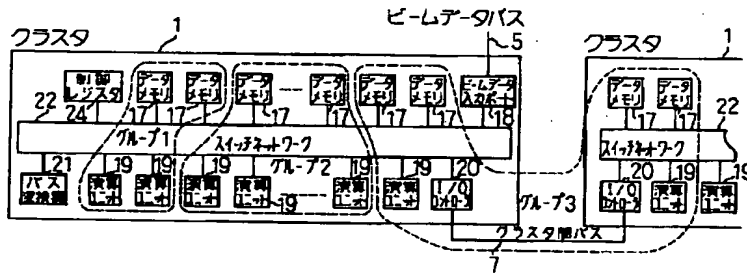
【図2】



【図5】



【図6】



【图8】

